Claims

1. A method of detecting invalid memory access used in a computer which executes a language system having a specific memory management function; a first program code that is executed under the control of the language system, and that accesses a first memory area reserved by the language system; and a second program code that is directly executed under the control of OS, and that accesses a second memory area reserved by the OS; wherein said method executed by the language system detects invalid memory access to the first memory area caused by the second program code, said method comprising the steps of:

allowing said language system to set the memory protection of the first memory area before the first program code calls the second program code;

calling and executing the second program code;

when a memory protection exception occurs, notifying of invalid memory access caused by the second program code to outside; and

when the execution of the second program code ends and the control returns to the language system, disabling the memory protection of the first memory area.

2. A method of detecting invalid memory access according to Claim 1, wherein:

when said memory protection exception occurs, if it is detected that the first program code performs normal memory access to the first memory area, said language system disables the memory protection to allow the normal memory access, and

5 then enables the memory protection again.

3. A method of detecting invalid memory access according to Claim 1, wherein:

if the first program code is executed under the multithread control, said language system suspends the

10 execution of other threads while a certain thread calls the second program code.

4. A method of detecting invalid memory access used in a computer which executes a language system having a specific memory management function; a first program code that is

15 executed under the control of the language system, and that accesses a first memory area reserved by the language system; and a second program code that is directly executed under the control of OS, and that accesses a second memory area reserved by the OS; wherein said method executed by the language system

20 detects invalid memory access to the first memory area caused by the second program code, said method comprising the steps of:

allowing said language system to save code information associated with the contents of the first memory area before

25 the first program code calls the second program code;

calling and executing the second program code;

when the execution of the second program code ends and the control returns to the language system, judging whether or not code information associated with the contents of the

5　first memory area coincides with the saved code information; and

if the code information associated with the contents of the first memory area does not coincide with the saved code information, notifying of invalid memory access caused by the

10　second program code to outside.

5. A method of detecting invalid memory access according to Claim 4, wherein:

when it is detected that while the second program code is called the first program code normally updates the first

15　memory area, said language system updates the saved code information based on code information associated with contents of the first memory area updated.

6. A method of detecting invalid memory access according to Claim 4, wherein:

20　if the first program code is executed under the multithread control, said language system suspends the execution of other threads while a certain thread calls the second program code.

7. A program used in a computer which executes a language

25　system having a specific memory management function; a first

program code that is executed under the control of the language

system, and that accesses a first memory area reserved by the

language system; and a second program code that is directly

executed under the control of OS, and that accesses a second

5   memory area reserved by the OS; said program allowing said

computer to execute language system's functions of detecting

invalid memory access to the first memory area caused by the

second program code;

wherein said computer executes the functions of:

10   setting the memory protection of the first memory area

before the first program code calls the second program code;

calling and executing the second program code;

when a memory protection exception occurs, notifying

of invalid memory access caused by the second program code to

15   outside; and

when the execution of the second program code ends and

the control returns to the language system, disabling the

memory protection of the first memory area.

8. A program according to Claim 7, allowing the computer

20   to execute the functions of:

when said memory protection exception occurs, if it is

detected that the first program code performs normal memory

access to the first memory area, disabling the memory

protection, allowing the normal memory access, and enabling

25   the memory protection again.

9. A program according to Claim 7, allowing the computer to execute the function of:

if the first program code is executed under the multithread control, suspending the execution of other threads while a certain thread calls the second program code.

10. A program used in a computer which executes a language system having a specific memory management function; a first program code that is executed under the control of the language system, and that accesses a first memory area reserved by the language system; and a second program code that is directly executed under the control of OS, and that accesses a second memory area reserved by the OS; said program allowing said computer to execute language system's functions of detecting invalid memory access to the first memory area caused by the second program code;

wherein said computer executes the functions of:

saving code information associated with the contents of the first memory area before the first program code calls the second program code;

calling and executing the second program code;

when the execution of the second program code ends and the control returns to the language system, judging whether or not code information associated with the contents of the first memory area coincides with the saved code information; and

if the code information associated with the contents of the first memory area does not coincide with the saved code information, notifying of invalid memory access caused by the second program code to outside.

5   11. A program according to Claim 10, allowing the computer to execute the functions of:

when said memory protection exception occurs, if it is detected that the first program code performs normal memory access to the first memory area, disabling the memory

10   protection, allowing the normal memory access, and enabling the memory protection again.

12. A program according to Claim 10, allowing the computer the function of:

if the first program code is executed under the

15   multithread control, suspending the execution of other threads while a certain thread calls the second program code.

13. A language system used in a computer which executes a language system having a specific memory management function; a first program code that is executed under the

20   control of the language system, and that accesses a first memory area reserved by the language system; and a second program code that is directly executed under the control of OS, and that accesses a second memory area reserved by the OS; wherein said language system detects invalid memory access to the first

25   memory area caused by the second program code, said language

system comprising:

means for setting memory protection of the first memory area before the first program code calls the second program code, for calling and executing the second program code, and

5 for notifying of invalid memory access caused by the second program code to outside when a memory protection exception occurs; and

means for disabling the memory protection when the execution of the second program code ends and the control

10 returns to the language system.

14. A language system according to Claim 13, further comprising:

means, when said memory protection exception occurs, if it is detected that the first program code performs normal

15 memory access to the first memory area, for disabling the memory protection, allowing the normal memory access, and then enabling the memory protection again.

15. A language system according to Claim 13, further comprising:

20 means, if the first program code is executed under the multithread control, for suspending the execution of the other threads while a certain thread calls the second program code.